Circuit design techniques for the high-performance CMOS IBM S/390 Parallel Enterprise Server G4 microprocessor

by L. Sigal
J. D. Warnock
B. W. Curran
Y. H. Chan
P. J. Camporese
M. D. Mayo
W. V. Huott
D. R. Knebel
C. T. Chuang
J. P. Eckhardt
P. T. Wu

This paper describes the circuit design techniques used for the IBM S/390® Parallel Enterprise Server G4 microprocessor to achieve operation up to 400 MHz. A judicious choice of process technology and concurrent top-down and bottom-up design approaches reduced risk and shortened the design time. The use of timing-driven synthesis/placement methodologies improved design turnaround time and chip timing. The combined use of static, dynamic, and self-resetting CMOS (SRCMOS) circuits facilitated the balancing of design time and performance return. The use of robust PLL design, floorplanning, and clock distribution minimized clock skew. Innovative latch designs permitted performance optimization without adding risk. Microarchitecture optimization and circuit innovations improved the performance of timing-critical macros. Full custom array design with extensive use of SRCMOS circuit techniques resulted in an on-chip L1 cache having 2.0-ns cycle time.

Introduction

The CMOS G4 central processor (CP) is a single-chip CMOS microprocessor that has been designed for the IBM S/390* Parallel Enterprise Server G4 system. It was designed to execute the S/390 instruction set at speeds up to 400 MHz. Achieving this performance required close integration of logic design and microarchitecture optimizations, high-performance CMOS circuit and array techniques, an advanced custom design tool suite, and full utilization of the advanced process technology. This paper describes the innovative CMOS circuit design techniques used to achieve high performance. The technology features and chip characteristics are discussed first,

©Copyright 1997 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/97/$5.00 © 1997 IBM

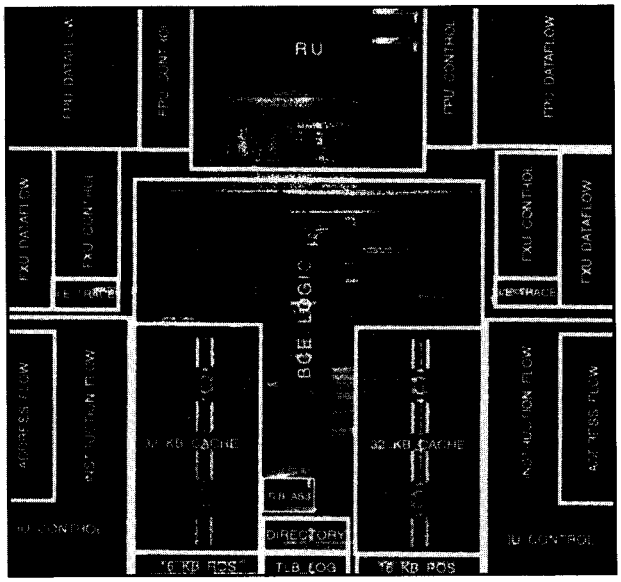

Figure 1

The CP chip micrograph (from [3], reproduced with permission; ©1997 IEEE).

Table 1 Typical chip technology parameters (from [3], reproduced with permission; © 1997 IEEE).

L_{eff}	0.2 μm
Gate oxide	5.5 nm
M1 pitch	1.2 μm
M2 pitch	1.8 μm
M3 pitch	1.8 μm
M4 pitch	1.8 μm
M5 pitch	4.8 μm
Power supply	2.5 V

Table 2 Key aspects of the CP chip (from [3], reproduced with permission; © 1997 IEEE).

Transistor count	7.8 million
Logic	3.8 million
Array	4.0 million
Die size	17.35 mm × 17.30 mm
Power dissipation	37 W @ 2.5 V 300 MHz
System bus frequency	Half of processor
On-chip decoupling cap	102 nF
C4 chip interconnections	1600
Off-chip signal I/O	448
Maximum frequency	400 MHz

followed by the chip floorplan and global design strategy. The phase-locked loop (PLL), which provides a processor clock that runs at twice the system bus frequency, is then described. High-frequency design issues for both the standard cell library and full custom circuits, the judicious choice and mixed use of static and dynamic circuits, and timing-driven placement are discussed. We then focus on the clock distribution and latch designs, which are critical for the high-frequency operation of all of the logic circuits. Examples are given for several timing-critical macros to illustrate how microstructure optimization and innovative circuit techniques are employed to improve timing. Finally, we discuss the array circuits, which use self-resetting CMOS (SRCMOS) circuit techniques extensively to achieve 2.0-ns access time and 500-MHz operation.

Technology features and chip characteristics

The microprocessor was implemented in the IBM CMOS 6S technology [1, 2]. Typical chip technology parameters are shown in **Table 1**. The technology features a 0.2-μm channel length, a 5.5-nm-thick gate oxide, low-resistance Ti-salicided n^+ and p^+ polysilicon and diffusions, shallow-trench isolation, metal fuses for laser cuts, n^+ precision resistors, five levels of metallization, and local interconnections. The power supply is 2.5 V.

A CP chip micrograph is shown in **Figure 1**. The chip measures 17.35 mm × 17.30 mm and contains 7.8 million transistors (**Table 2**). A phase-locked loop provides a processor clock that runs at twice the system bus frequency. There are about 3.8 million logic transistors and 4.0 million array transistors on the chip. The measured power dissipation at 300 MHz is 37 W. The chip contains 1600 C4 terminals and 448 off-chip signal I/Os, and has operated successfully at frequencies up to 400 MHz.

Dedicated thin-oxide capacitors totaling 102 nF are provided for on-chip decoupling. This, combined with the inherent nonswitching well-to-substrate and diffusion-to-well capacitances, provides about 200 nF of on-chip decoupling capacitance. Each capacitor has a gated n-FET control device with an external decap-enable pin for leakage current measurements during test. Additionally, each capacitor contains a "built-in" fuse which, in the presence of a large current resulting from oxide defects, results in an open in the first level of chip metallization (M1). The decoupling capacitor cell (**Figure 2**) is designed to fit under the dataflow wiring tracks. The cell is double-bit-pitch wide (43.2 μm) and 14 tracks tall (25.2 μm). Two of the fourteen horizontal wiring tracks are specifically blocked for the decoupling capacitor wiring so that the capacitor can fit under the wiring. A low-resistance layout of the capacitor cell provides a fast time constant, approximately 85 ps.

Chip floorplan and global design strategy

The CP chip comprises five major units: instruction unit (IU), fixed-point unit (FXU), floating-point unit (FPU), buffer control element (BCE), and recovery and error-

detection unit (RU) [3]. The IU, FXU, and FPU are duplicated on the chip for increased system reliability. The BCE and RU, with error-detection logic, are centrally located in the floorplan to support communication with both sets of instruction and execution units. The BCE contains a unified 64KB L1 cache [4] and 32KB read-only memory (ROM) for millicode storage. The RU maintains an error-checking and correction (ECC) protected copy of the processor states, and various system support functions for error detection and recovery. The PLL is located near the center of the chip and generates internal system clocks that run at twice the system bus frequency.

Each unit consists of many smaller design elements called macros, and within each unit macros are classified as dataflow or control. Dataflow macros are custom-designed and are arranged in bit stacks. Most of the dataflow and control logic was implemented with static CMOS circuits. Dynamic circuits were used only in isolated, performance-critical cases. Approximately 75% of the logic macros were implemented with custom schematics and layouts, and the remaining 25% were synthesized.

Control macros (also designated as random logic macros, or RLMs) are synthesized logic. This logic was implemented using a standard-cell approach in which gates are selected from a common fixed library and are placed and routed with automated tools. In most cases the RLMs also included gate array backfill of spare logic in the unused area within the macro. The gate array's spare logic consists of unwired transistors which can be personalized at the metal-only levels to facilitate quick design changes.

Functional descriptions in VHSIC Hardware Description Language (VHDL) and physical design hierarchies match at the macro level. The physical partitioning of the chip was largely influenced by the logical partition at the macro level. The chip physically comprises at least three levels of hierarchy (chip, unit, and macro), and design implementations were achieved concurrently at each level. Thus, interconnections between units and interconnections between macros were implemented in parallel with the macro layouts. Our concurrent design approach supported both a top-down and bottom-up design methodology, with constraints being passed at each level of the hierarchy. The shape and size of arrays and width of dataflow stacks were designed from the bottom up. The aspect ratio of control macros and placement of control pins were designed from the top down. Design constraints were passed in both directions of the hierarchy in the implementation of the chip power distribution. The power mesh on higher metal levels (M4 and M5) was predefined at the chip level and considered fixed at the unit and macro levels. The power mesh on the M3 level and below was designed from the bottom up, giving the most flexibility to the macros and units. In some cases, such as over the dataflow areas, the M2 power mesh was predefined.

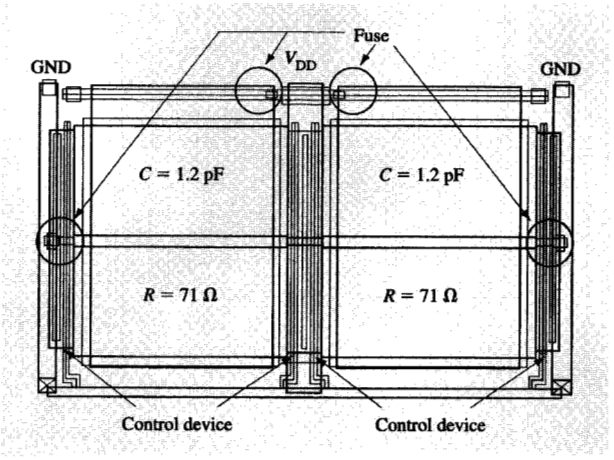

Figure 2

Decoupling capacitor layout (from [3], reproduced with permission; ©1997 IEEE).

The power distribution supports an average dc voltage drop of 23 mV. The current transients were managed by including additional on-chip decoupling capacitors around large noise sources such as the off-chip drivers, clock buffers, and on-chip drivers with large loads. Since a large amount of switching capacitance occurs in the dataflow stacks, decoupling capacitors were also placed under the wiring tracks with little or no area impact.

Phase-locked loop (PLL)

The PLL provides a processor clock that runs at twice the system bus frequency. It has a current-controlled oscillator and an on-chip chip-generated analog V_{DD}. It operates over the range from 36 MHz to 571 MHz, with less than 4 ps/mV of long-term phase error between the PLL output and reference clock due to power supply noise, and less than a 1-ps/mV reduction in cycle time due to other noise sources. **Figure 3** shows a block diagram of the PLL. The phase detector generates pulses with widths equal to the phase difference between its two inputs (the reference clock f_{ref} and $f_{clock}/M_{feedback}$, where f_{clock} is the chip clock). The charge pump converts the pulse width into charge stored on a capacitor pair to set a differential bias. The amount of charge on the capacitor is increased or decreased until the pulse widths are reduced to the minimum value (i.e., no phase difference between f_{ref} and $f_{clock}/M_{feedback}$), and the correct bias voltage is reached. The voltage-to-current converter converts the bias voltage to a current which is injected into the current-controlled

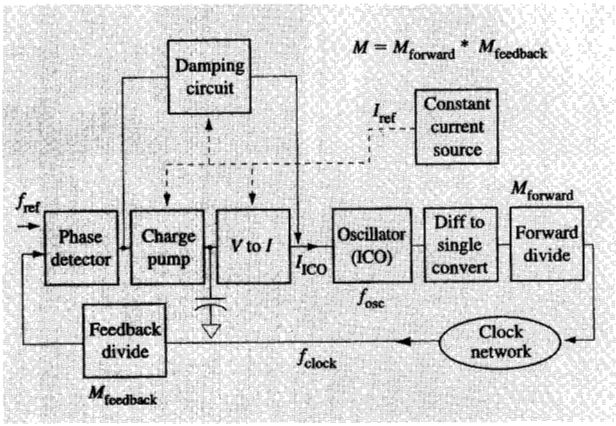

Figure 3

Block diagram of PLL.

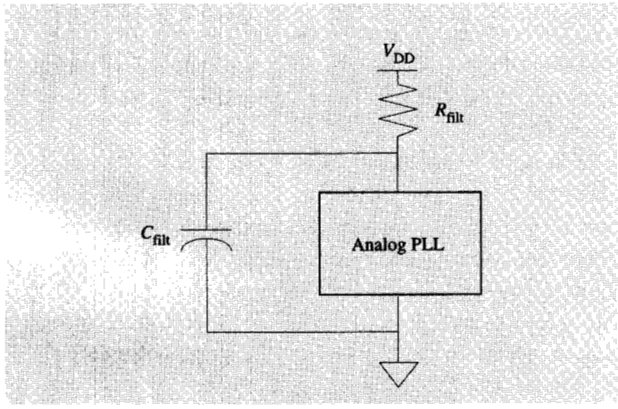

Figure 4

On-chip analog V_{DD} generation with *RC* filter.

oscillator (ICO). The amount of current determines the oscillator frequency. The forward divide and feedback divide determine the relationship between the frequencies of the reference clock and the oscillator and chip clock, such that $f_{clock} = f_{osc}/M_{forward}$ and $f_{ref} = f_{clock}/M_{feedback}$, where $M_{forward}$ and $M_{feedback}$ are the divide values of the forward and feedback divides, respectively. A constant-current source generates the reference current I_{ref}, which biases the current-controlled analog circuits of the PLL.

A primary barometer of PLL performance is jitter. In the CP, the PLL aligns its clock with other chips in the system. Consequently, accumulated phase error (the phase difference between the PLL output and the reference clock) is important. Other critical measurements are cycle-to-cycle jitter and long-term jitter, which reduce the cycle period available for logic operations. In the present design, there are three places where PLL performance can be severely degraded by power supply noise—the constant-current source (I_{ref}), the current injected into the ICO by the voltage-to-current converter, and the ICO itself. The I_{ref} supplies the constant current which biases the entire PLL. Any fluctuation in the current provided results in both cycle-to-cycle and long-term jitter. This error is reduced by minimizing the sensitivity to supply voltage (dc shifts and low-frequency noise), and by the use of capacitive coupling (high-frequency noise). The voltage-to-current converter supplies bias current to the oscillator. Noise added to this bias results in both types of jitter. The output of the converter is single-ended and can be sensitive to power-supply shifts. The present design not only minimizes the injected noise, but also utilizes common-mode noise-rejection technique so that the noises are added as common mode and cancel out at the output. The ICO is a differential cascoded current source design which relies on the variation of the input current to control the frequency.

To minimize the noise and add flexibility to the PLL usage, the analog V_{DD} is generated on-chip with the simple *RC* filter shown in **Figure 4**. This approach eliminates the need for an additional power supply and the associated package-coupled noise, and makes the design compatible with the existing power-supply systems. The on-chip coupling noise on the analog V_{DD} is minimized both by careful placement and wiring and by the low-impedance path to V_{DD} through the *RC* filter.

High-frequency design strategy

The fundamental design strategy was to design the dataflow to the most aggressive cycle time achievable, and then to match the control portion to the dataflow cycle time. In cases where control paths lag the dataflow cycle time after all possible logic optimizations, cycles per instruction (CPI) were traded off for cycle time. The CP achieved its high-frequency design goals by unique combinations of improved timing methodologies and design techniques. These methodologies included advanced synthesis techniques for automatically generating timing-optimized schematics for the control functions, and more accurate extraction and modeling of the layout capacitances and resistances. Two types of high-frequency design strategies (logical and physical) were developed to resolve long-path timing problems. The first type incorporated high-leverage logic design optimizations, and included macro-partitioning optimizations of the timing-critical logic and VHDL optimizations [5]. When logical strategies did not resolve a long-path timing problem, physical strategies of placement, wiring, high-performance circuit, and clocking techniques were exercised. Most

dataflow macros were implemented with custom static circuits. All control macros were implemented from a static cell library with fine power granularity.

- *Standard cell library*

A completely full custom implementation was not feasible because of the complexity of the S/390 architecture and our resource and schedule constraints. Thus, two carefully tuned standard cell booksets were developed for use with synthesis. One of these contained fixed schematics and layouts, and the other was parameterizable with capabilities for automatic layout generation [5]. Both bookset designs were limited to relatively simple functions: INVERTERs and BUFFERs; two-, three-, and four-way NANDs and ANDs; two- and three-way NORs and ORs; and simple AOIs, OAIs, AOs, and OAs (2 by 1 and 2 by 2). Many nonbuffered power levels were designed for each book type. These power levels were created both by reducing the width of the FET fingers and by adding FET fingers. Within the fixed bookset, each book type was individually simulated and beta-tuned (ratio of p-FET device width to n-FET device width) to achieve nearly equal rising and falling output delays.

The parameterized bookset offered the most flexibility and highest performance. Multiple beta ratio books were designed for the book types with the highest usage. The beta ratio was changed by both increasing and decreasing the ratio of p-FET width to n-FET width. Thus, rising input delay was improved to speed up the falling output delay, and vice versa. (Synthesis exploited these various beta ratios as it attempted to minimize both falling-input-to-falling-output delay and rising-input-to-rising-output delay for a path.) All of the books in this two-dimensional bookset (the two dimensions being power level and beta ratio) were timing-characterized for synthesis. Timing characterization included input-pin-to-output-pin base delays, and delay sensitivities to input-pin slew and output-pin capacitive load. Synthesis using these carefully tuned booksets yielded schematics which were nearly as optimal as full custom schematics.

Much effort was expended to minimize the internal parasitic resistances and capacitances of each book. Polygate resistances were reduced by placing the input pin between the n-FET and p-FET fingers. Source and drain diffusion capacitances were reduced by sharing as much of the diffusion as possible between adjacent-series-connected p-FETs and n-FETs. The local interconnection level was leveraged for the internal book connections of FET drains and sources.

- *Full custom circuit design*

Each custom circuit designer had full control of circuit schematics and circuit topologies, and this permitted them to exploit fully the delay advantages of transmission gates, AOIs, and OAIs. Transmission gates were used primarily to implement selector functions when the data path was more timing-critical than the select path. Manual logic transforms were applied to the schematics to reduce the number of stages and critical path delays through the schematics. Transforms included buffer insertion and deletion, NAND–NAND-to-AOI conversions, NOR–NOR-to-OAI conversions, and Shannon expansions. The noncritical loads on a timing-critical net were buffered in a manner which reduced the load on the net's driver.

After the circuit schematic optimizations were completed, individual gates were tuned. Gates in the non-timing-critical paths were resized to achieve 10–90% rising and falling slew targets [5]. Next, stage gains in the critical paths were reduced to a value typically between 2 and 3, and equalized throughout the entire path. Stage gain was defined to be the ratio of a gate output load capacitance to its input pin capacitance. The output drivers (gates driving the macro output pins) were properly sized to drive the full lumped load capacitance instead of buffering the signal via multiple inverters to increase drive strength. The preceding gates (i.e., gates driving output drivers) were accordingly sized up to reduce their stage gain. Stage gains were also reduced by increasing the FET widths of the critical gate load on a timing-critical net and/or by lowering the FET widths of the noncritical gate loads on a timing-critical net. Equalizing stage gains was accomplished by increasing and/or decreasing the FET widths in the path until the slews of all of the nets in that path were approximately equal. (Excessive gain on a circuit stage causes an excessive slew on the output net of that stage.) The beta ratios of books were modified to equalize the rising-input-to-rising-output delay and falling-input-to-falling-output delay for the timing-critical paths, as synthesis did for the control functions.

The layout techniques employed during development of the standard cell booksets were also used during full custom circuit layouts. Additionally, to minimize wire capacitance in timing-critical custom paths, circuits were manually placed and wired. In situations where long wires were unavoidable (principally on nets with large fanout), wire *RC* delay was modeled with various wire widths to find the width which resulted in minimum path delay.

- *Dynamic circuits*

Array circuits such as the L1 cache, L1 directory, TLBs, ROM, traces, and write buffer were designed with self-resetting CMOS (SRCMOS) circuits [4]. The only nonarray macros designed with dynamic circuits were the cache hit logic [6], the dynamic multiplexors used with latches, and the divide and square-root lookup ROM in the FPU. The rest of the macros were designed with static circuits. All known static techniques were applied before considering dynamic circuit solutions.

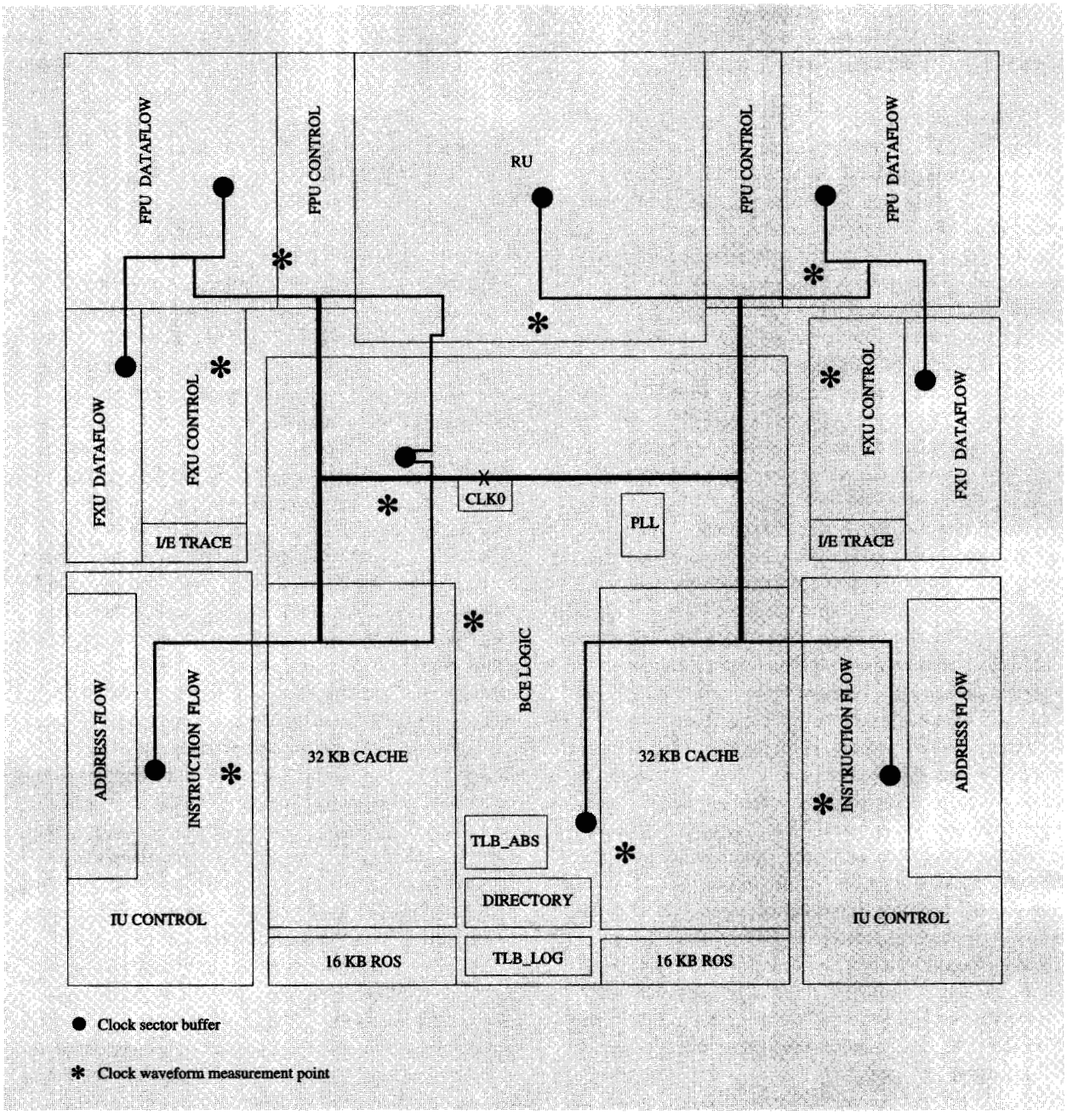

Figure 5

Chip floorplan and clock distribution (from [3], reproduced with permission; ©1997 IEEE).

• *Timing considerations*

The address-generation path was still timing-critical after application of all of the physical placement, wiring, and circuit techniques described above. Fortunately, the L1 cache and ROM access paths had been designed with timing margin; thus, some of this margin was transferred to the address-generation path. This was accomplished by simply adding a programmable clock delay block within the cache and ROM arrays. In other similar situations, where a path was timing-critical and the prior or

subsequent cycle path had timing margin, we opted to reposition the latch to rebalance the delays of the back-to-back cycle paths [5].

Many critical paths arose because of excessive delay in the propagation of a signal between distant macros. The microprocessor floorplan was modified to move these macros closer together; even small movements improved timing, since (to first order) wire propagation delay is proportional to the square of wire length. If the wire was still timing-critical, several other solutions were explored. For wires with more than one load, we investigated the feasibility and timing impact of duplicating the net driver and net so that one of the drivers merely drives the timing-critical load. Solutions involving wide or tapered wires were analyzed via AS/X simulations. Widening a wire reduces its resistance at a faster rate than its capacitance increases, thus permitting a reduction in the *RC* delay of the wire. A tapering solution involved the use of a wider section of wire at the near (driver) end and a narrower section of wire at the far (receiver) end. These solutions further reduced its *RC* delay. Care was taken to ensure that no polysilicon wires were used on critical wiring nets.

In addition to implementing techniques to reduce cycle times, careful noise analysis was required to ensure that capacitive metal-to-metal noise couplings would not adversely affect the functionality of the circuits at high frequencies. All distant inputs to transfer gate latches were buffered locally with receivers. The parasitic coupling analysis tools identified nets that required grounded metal shields.

- *Design for reliability*

It was challenging to design this high-speed microprocessor to operate functionally in an accelerated burn-in environment. In normal operation, the CP operates in a 2.5-V environment. However, to ensure high quality and reliability, the CP was required to pass high-voltage and temperature stress testing. A dynamic voltage screen (DVS) test consists of writing patterns at 4.25 V, while a 5.0-V bump is applied at the end of each DVS pattern. The CP was designed to be functional and pass I_{DDQ} leakage currents under these stress conditions. In addition, the CP was also designed to be functional under burn-in conditions at 140°C and 3.75 V.

Clock distribution and latches

The microprocessor clock is a single-phase clock that is distributed from the chip central clock buffer to all latches inside the macros in three levels of hierarchy. The first two levels of distribution are in the form of balanced H-like trees. The first-level tree routes the global clock from the central clock buffer to the nine sector buffers. Each IU, FXU, FPU, and RU unit has one sector buffer, while the BCE has two sector buffers. **Figure 5** shows the first-level clock wiring tree. The sector buffers repower the clock to all macros inside the sectors. There are 580 macro clock pins among all the units. The clock propagation delay along the tree is balanced against the macro input capacitance and *RLC* characteristics of the tree wires. To form the horizontal wiring of each tree, use is made of the relatively low-resistance M5 layer. At various places along the tree, inductive coupling is reduced, and the return path is improved by using power wires for shielding. Decoupling capacitors are incorporated into central and sector buffers to reduce delta-I noise. Extensive AS/X circuit simulations were done to guarantee low clock skew at macro pins. Typical simulated *RLC* delay of the first-level tree is 300 ps, with a skew of 20 ps at the sector buffers. The sector buffer delay is 230 ps. Typical simulated *RLC* delay within sectors is 210 ps, with a skew of 30 ps at the macros. **Figure 6** shows the measured waveforms of the central clock buffer output and clocks at ten points (marked on Figure 5) of the 580 macro pin locations driven by the second-level clock tree. The results indicate a mean delay of 740 ps and less than 30-ps skew from the central clock buffer to the macro pin. The last level of clock distribution is local to each macro. **Figure 7** shows the clocking scheme within macros. From the macro pin the clocks are wired to the clock blocks. The overall target skew for this wiring is less than 20 ps. For large-area macros, multiple clock pins were used to reduce the wire

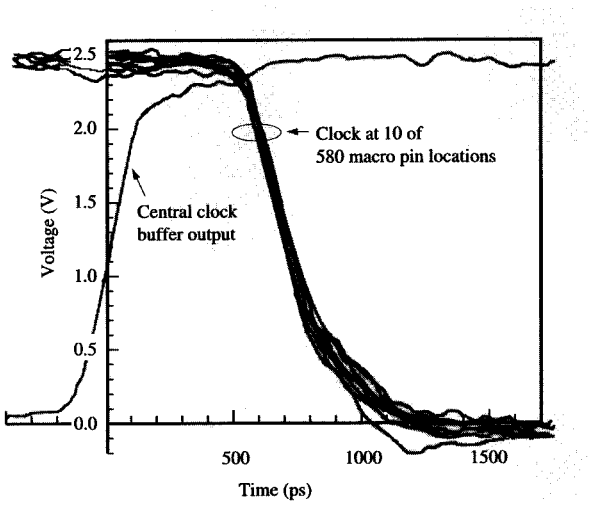

Figure 6

Measured clock waveforms at ten points of the 580 macro pin locations indicated on Figure 5 (from [3], reproduced with permission; ©1997 IEEE).

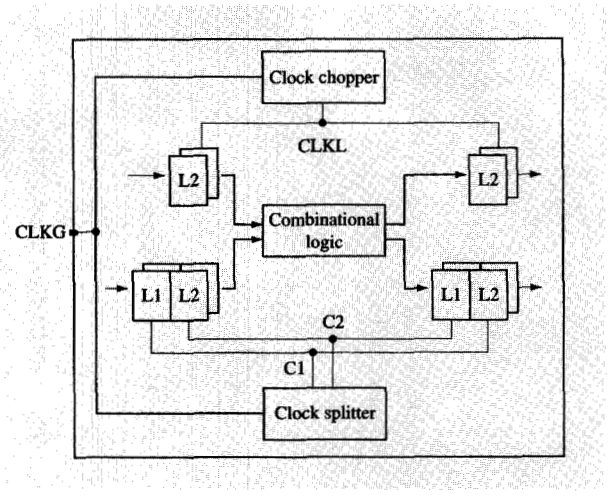

Figure 7

Clocking scheme within macros.

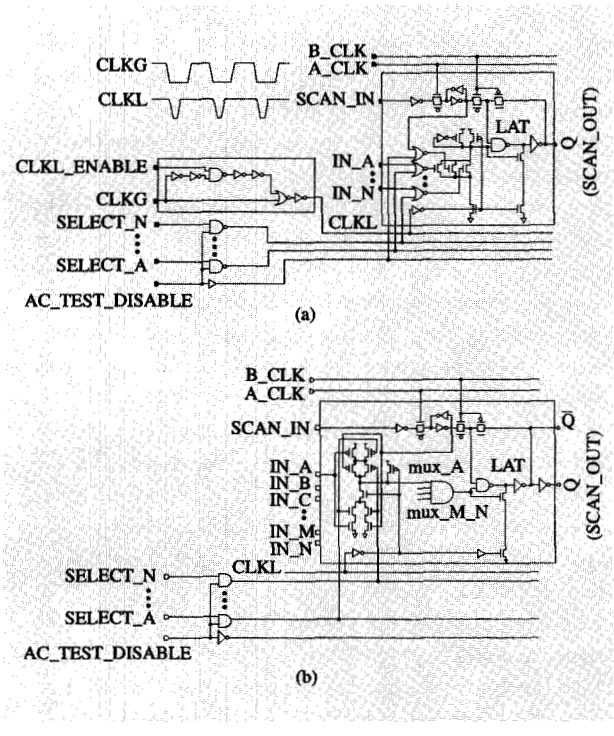

Figure 8

(a) Clock block/latch combination: generation of local clock 'CLKL' and latch with dynamic multiplexor (from [3], reproduced with permission; ©1997 IEEE). (b) Clock block/latch combination: latch with preset static multiplexor (from [3], reproduced with permission; ©1997 IEEE).

length to clock blocks. The clock block generates local clocks that drive latches, as is explained later. The target skew for local clocks is less than 50 ps. All macro-level wiring is done by hand for custom macros or with a place-and-route tool for synthesized macros. For synthesized macros that contained many latches, and therefore multiple clock blocks, a clock optimization tool was used that reassigned latches to clock blocks on the basis of placement. This resulted in clock blocks driving latches that were placed closest to them. Macro layouts were extracted for R and C parasitics, and the extracted netlists were used to time the macros. Therefore, any skew in the last level of clock distribution was captured in that macro's timing abstraction.

Two types of latches are used on the microprocessor outside the arrays: L2-only latch and L1–L2 pair; both are cycle boundary latches (i.e., there are no midcycle or phase latches). The cycle boundary is defined to occur at 'CLKG' falling. Corresponding to the two latch types are two types of clock blocks. The first clock block/latch combination is shown in **Figure 8(a)**. This clock block chops the global clock on the falling edge to create a short pulse 'CLKL' that triggers the latch. By using either a dynamic multiplexor in front of the latch [Figure 8(a)] or a preset static multiplexor [**Figure 8(b)**], a fast latch delay is achieved. This mux/latch combination interfaces smoothly with the static circuits, yet allows fast delays for multi-input high-fanout registers typical of the dataflow. While 'CLKL' is inactive (high), the dynamic multiplexor is in the precharge state while the latch is holding its state. Similarly, the static multiplexor is preset high (node mux_A is high) while the latch is holding its state. When the 'CLKL' low-going pulse arrives, the latch node 'LAT' begins to discharge. If the multiplexors' data/selects are such that the multiplexors remain in their precharged/preset state, node 'LAT' discharges fully. If, however, the multiplexors evaluate, node 'LAT' is driven high. The n/p transistors in the multiplexors are skewed to favor the transition launched by the arrival of the 'CLKL' pulse. Strength ratios in the preset static multiplexor are limited to 2.5:1 to maintain a reasonable noise margin and to allow adequate time for presetting all gates at the end of the clock pulse. The seven-input preset static multiplexor evaluates in about 225 ps compared to about 400 ps for a standard static multiplexor with the same input capacitance and area. When 'CLKL' is active, the latch is in the transparent mode. Full chip and macro wire RC extractions were carried out to guarantee no early-mode problems.

The second clock block/latch combination is shown in **Figure 9**. This clock block splits the global clock on the falling edge to create C1/C2 clocks. C1–C2 clock overlap at the cycle boundary is set close to 0 ps. Having a positive overlap would reduce the latch propagation delay,

but it would also require more early-mode padding. These latches are used in non-timing-critical dataflow macros and in control macros where all latches are single-input and the speed advantage of an L2-only latch is reduced.

All of the latches used in the design are fully scannable and LSSD-compatible. The largest area penalty for fully scannable latches was in the general-purpose register (GPR) arrays at 25%. The overall chip area penalty is less than 5%. Having an LSSD-compatible design increased productivity during test vector generation and allowed greater than 99.5% dc stuck-at-fault test coverage. The goal of the design was also to achieve high ac transition test coverage, since ac test coverage often suffers from an inability to create appropriate transitions due to latch adjacency in the scan chain. This is most prevalent in dataflow macros where complicated logic functions are being done (e.g., multipliers). To eliminate the latch adjacency problems, nonfunctional scan-only latches were inserted into some macros for every register bit. With this addition, ac transition test coverage greater than 91% was achieved.

Specialized circuitry was added to all clock blocks to allow for edge shifting at the cycle boundary. Taking the clock splitter as an example, we are able to

- Completely un-overlap C1 falling and C2 rising by a large amount. This provides a work-around for potential early-mode problems.
- Delay C1 falling from its nominal value. This allows stressing of early mode and provides a determination of how much margin exists in the design.
- Delay C1 falling and C2 rising together from their nominal values. This allows cycle stealing from the previous cycle.

The last feature was found to be extremely useful in debugging late-mode problems. Since clock stressing is local to each macro, when the first late-mode path was found, more time was given to the failing cycle through cycle stealing. This usually did not create late-mode problems in the following cycle for the stressed macro. With the worst late-mode path now masked, work could continue in finding the next late-mode path that failed. **Figure 10** shows how the clock splitter block was modified to cause un-overlap between C1 falling and C2 rising.

Microarchitecture optimization and circuit innovations

In this section, we present several examples of how microarchitecture optimization and innovative circuit techniques were employed to improve timing.

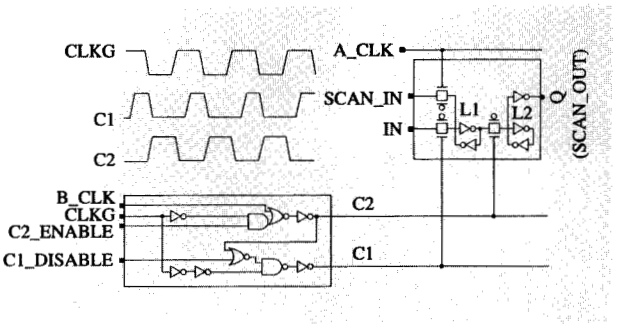

Figure 9

Clock block/latch combination: generation of C1/C2 clocks for L1/L2 latches (from [3], reproduced with permission; ©1997 IEEE).

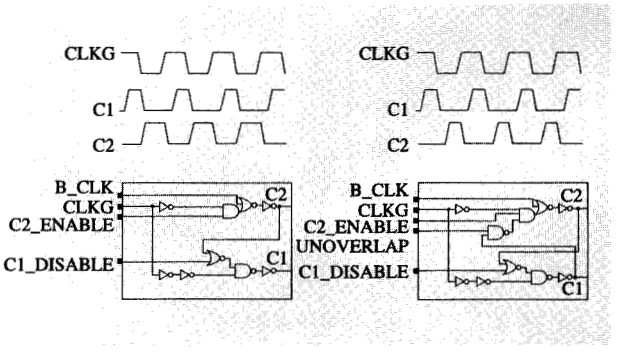

Figure 10

Use of clock splitter/clock block to separate C1 falling edge and C2 rising edge.

- *FXU binary-zero-detect circuit*

The execution of instructions in the fixed-point unit involves computing results in the 64-bit binary adder, 32-bit decimal adder, and 64-bit logic unit, and setting the condition codes—all in a single cycle. The critical path was computing the 64-bit binary sum, doing a 64-bit zero-detect on the sum, and using the result to set binary instruction condition codes. A scheme was used where the 64-bit zero-detect was initiated and completed before the binary sum result was actually available [7]. This led to an execution cycle of less than 3 ns.

- *FPU 120-bit adder*

The 120-bit floating-point adder was implemented as a carry-select adder. Care was taken to make sure that the adder would never produce a negative result. This

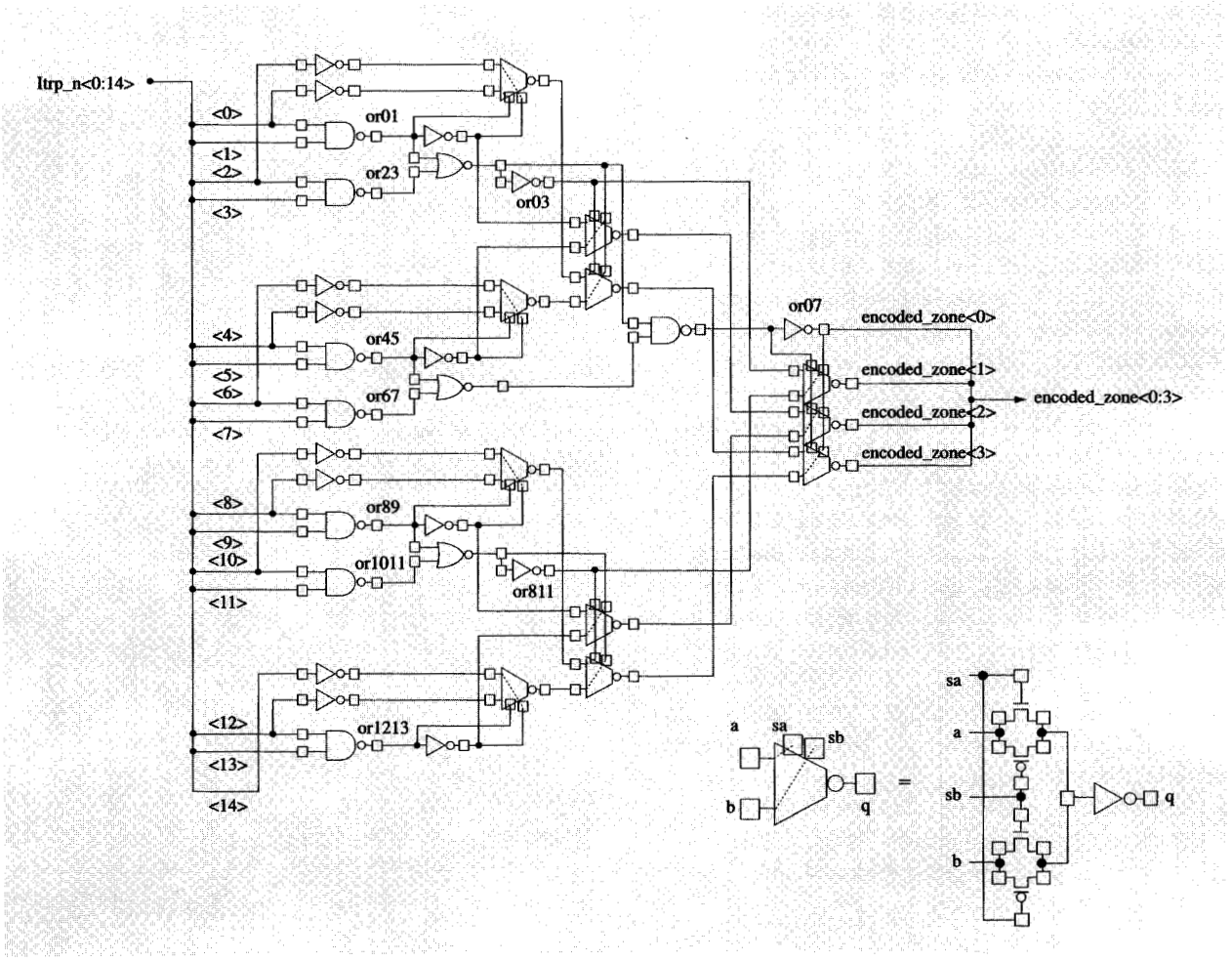

Figure 11

Fast priority encoder circuit with pass-gate multiplexor.

eliminated the step of converting a two's-complement negative result to sign-magnitude format. The adder was split into 15 bytes, and each byte generated two sums. The critical path was the carry-merge tree, which was implemented in a binary tree fashion. This carry-merge tree produced the 15 selects for byte sums. The adder computed the result in 1.35 ns in eleven levels of logic.

• *Binary priority encoders*

Several places in the logic design required the use of priority encoders. An example was a priority encoder in asynchronous interrupt logic where a 16-bit asynchronous interrupt vector had to be binary-encoded. Another use of this circuit was in leading-zero detectors. A fast priority encoder circuit was designed with pass-gate multiplexors, as shown in **Figure 11**. This circuit computes the result in about five levels of logic on 16-bit input vectors.

• *FXU 64-bit rotator*

A 64-bit rotator in the fixed-point unit was implemented in two levels of 8-to-1 multiplexors. A straightforward implementation would look like that shown in **Figure 12(a)**. In this implementation bits 0–6 of the first-level 8-to-1 multiplexor had to drive the entire width of the data stack to reach bits 57–63. This increased the wire capacitance of these bits eightfold and significantly increased the delay of the first-level multiplexor. The implementation that was used instead duplicated some bits of the first-level multiplexor shown in **Figure 12(b)**. This increased the capacitance of the first-level multiplexor select and data lines by only 12.5%, while keeping the

First level 8:1 multiplexor for bitX:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63
8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	..	0	1	2	3	4	5	6	7
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	..	8	9	10	11	12	13	14	15
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	..	16	17	18	19	20	21	22	23
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	..	24	25	26	27	28	29	30	31
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	..	32	33	34	35	36	37	38	39
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	..	40	41	42	43	44	45	46	47
56	57	58	59	60	61	62	63	0	1	2	3	4	5	6	7	..	48	49	50	51	52	53	54	55

Second level 8:1 multiplexor for bitX:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	..	57	58	59	60	61	62	63	0
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	..	58	59	60	61	62	63	0	1
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	..	59	60	61	62	63	0	1	2
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	..	60	61	62	63	0	1	2	3
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	..	61	62	63	0	1	2	3	4
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	..	62	63	0	1	2	3	4	5
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	..	63	0	1	2	3	4	5	6

(a)

First level 8:1 multiplexor for bitX:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63	0	1	2	3	4	5	6
8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	..	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	..	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	..	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	..	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	..	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	..	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
56	57	58	59	60	61	62	63	0	1	2	3	4	5	6	7	..	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62

Second level 8:1 multiplexor for bitX:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..	56	57	58	59	60	61	62	63							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	..	57	58	59	60	61	62	63	64							
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	..	58	59	60	61	62	63	64	65							
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	..	59	60	61	62	63	64	65	66							
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	..	60	61	62	63	64	65	66	67							
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	..	61	62	63	64	65	66	67	68							
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	..	62	63	64	65	66	67	68	69							
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	..	63	64	65	66	67	68	69	70							

(b)

Figure 12

FXU 64-bit rotator implemented in two levels of 8-to-1 multiplexors: (a) straightforward implementation; (b) improved implementation where some bits of the first-level multiplexor are duplicated.

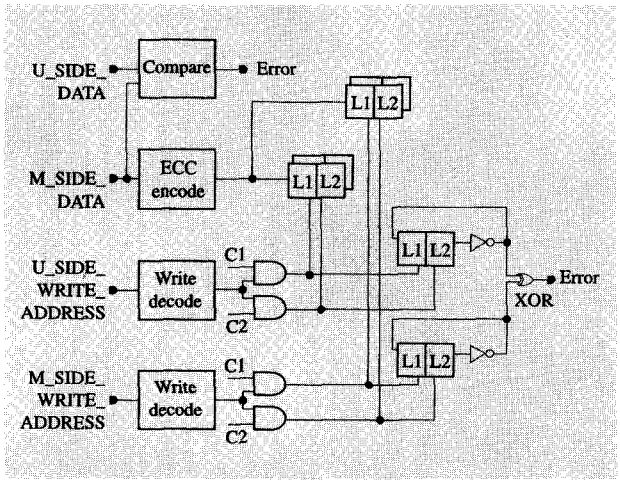

Figure 13

Clock-checking scheme for writing data into RU checkpoint array.

wire capacitance of the second-level multiplexor small. We found the latter implementation to be much faster.

- *Interleaving bits in the FXU data stack*

The fixed-point-unit data stack was 64 bits wide to accommodate instructions that work on 64-bit operands. However, a large number of instructions in the architecture move data from the upper 32 bits to the lower 32 bits and vice versa. To minimize both area and cycle time, we avoided the many wiring channels that would be required for this by interleaving bits 0, 32, 1, 33, $\cdots$, 31, 63. This had zero cost on most macros that are bit-sliced, such as registers, GPR, FPR, mask, and BLU. Even in the 64-bit rotator there was no penalty. The only macro that was made more complicated was the 64-bit binary adder in its carry-merge tree.

- *Clock checking in the RU unit*

Every cycle datum that represents the architected state of the processor is sent to the RU unit from both copies of the processor to be compared, ECC-encoded, and written into the checkpoint array. Special precautions had to be taken to ensure that data are written into the checkpoint array at the appropriate time. There can be a malfunction in the write address decode circuit or the clock circuit that can prevent the clock from becoming active (thus not writing the checkpoint array), or cause it to become active when it should not, causing a spurious write. Therefore, the entire data vector cannot be written into the checkpoint array by a single clock—two clocks are required so that they can check each other. The data vector is partitioned into two halves, and a unique clock is sent to each half, as shown in **Figure 13**. The use of two toggle latches with an error detector guarantees that a single fault in the write address decode circuit or clock circuit will be detected.

Array circuits

The CP has a unified 64KB store-through L1 cache. It uses a line size of 128 bytes to buffer the data needed by the processor units as well as to interface with the off-chip L2 cache. The L1 cache was an absolute address cache with four-way set associativity and two-way interleaving. Because of the store-through design, the L1 uses parity instead of ECC while maintaining good error recoverability. During a cache fetch operation, the L1 cache, directory, and TLB are read simultaneously in the beginning of the cycle. Results from the directory and TLB are fed into the address comparators and encode logic to determine whether a cache hit occurs. If a hit occurs, a 1-out-of-4 late-select signal is generated and passed to the cache output multiplexor to select one of the four cache sets. The directory, TLB, L1 cache, and the associated comparator and hit decode logic must all operate within a single system cycle.

To achieve these performance goals, the arrays utilized various innovative dynamic circuit techniques. Some of the key design attributes are

- A six-device CMOS SRAM array cell (33-μm^2 cell area).
- A high-density ROM array cell (4.4-μm^2 cell area).
- Multiport register file cells.
- Peripheral array circuits utilizing synchronized pipelined designs with dynamic and self-resetting circuit techniques for fast access and cycle times [4].
- Array circuits (all of them) designed for system robustness to be fully functional at extended power supply and temperature ranges with worst-case process parameters.
- Advanced ABIST (array built-in self-test) engines. These are used in the SRAM and ROM macros to provide comprehensive array test coverage. The ABIST engines are scan-programmable for pattern-generation flexibility and are capable of performing accurate cycle- and access-time measurements.

Table 3 lists the characteristics of various high-performance array macros. As the table shows, these arrays achieved bipolar-like access and cycle times with the density leverage of CMOS technology. The L1 cache achieved 2.0-ns access and operation up to 500 MHz.

Summary

We have described the circuit design techniques used for the S/390 G4 microprocessor to achieve operation up to

Table 3 Characteristics of various array macros.

Array type	*Macro density* (Kb)	*Macros per chip*	*Access/Cycle* (ns)	*Power* (mW/Kb)	*Area* (Kb/mm^2)
L1 directory	15	1	<1.5/2.5	60	7
TLB-A	7	1	<1.5/2.5	60	7
TLB-L	14	1	<1.5/2.5	60	7
L1 cache	288	2	<2.4/2.5	13	14
ROS	144	2	<2.5/2.5	6	62
Store buffer	4.5	1	<1.4/2.5	58	3

400 MHz. The target performance of 300 MHz was exceeded by focusing on timing and high-frequency circuit design methodology and techniques. At the architecture level, cycle time was emphasized over cycles per instruction. Both synthesis and placement methodologies were developed with timing as top priority. PLL design, chip floorplan, and clock distribution were skew-optimized. For the custom design suite, judicious mixed use of static, dynamic, and SRCMOS array circuits achieved high-frequency operation without sacrificing design time. Innovative latch designs allowed performance optimizations in a controlled manner. Microarchitecture optimizations and circuit innovations were shown to be effective in improving the performance of timing-critical macros. The well-planned concurrent top-down and bottom-up design approach has been shown to be viable for high-frequency designs as well as for reducing design risk and shortening the design time.

Acknowledgments

The authors would like to acknowledge the contributions of the following persons: R. Allen, C. Anderson, R. Averill, J. Badar, D. Bair, A. Barish, U. Bakhru, D. Balazich, K. Barkley, D. Beece, D. Becker, I. Bendrihem, M. Billeci, F. Bozso, J. Braun, T. Bucelot, J. Burns, C. Bui, S. Carey, Y. Chan, M. Check, C. Chen, K. Chin, E. Cho, J. Clabes, S. Coleman, R. Crea, A. Dansky, J. Dickerson, R. Dennis, G. Ditlow, R. Dussault, P. Emma, K. Eng, R. Farrell, R. Franch, J. Feldman, B. Giamei, J. Gilligan, B. Grossman, A. Gruodis, H. Hansen, R. Hanson, A. Hartstein, R. Hatch, D. Heidel, D. Hillerud, D. Hoffman, D. Hui, K. Jenkins, J. Ji, D. Jones, G. Jordy, G. Katopis, V. Klimanis, A. Kohli, N. Kollesar, T. Koprowski, S. Kowalczyk, B. Krumm, C. Krygowski, L. Lacey, L. Lange, K. Lewis, W. Li, J. Liptay, P. Liu, P. Lu, J. Ludwig, V. Lund, R. Mauri, S. McCabe, T. McNamara, T. McPherson, D. Merrill, P. Minear, P. Muench, M. Mullen, J. Navarro, J. Neely, H. Ngo, T. Nguyen, G. Northrop, D. Ostapko, P. Patel, A. Pelella, E. Pell, C. Price, J. Rawlins, W. Reohr, P. Restle, S. Risch, R. Rizzolo, B. Robbins, R. Robortaccio, G. Scharff, M. Scheuerman, E. Schwartz, A. Sharif, K. Shepard, K. Shum, T. Slegel, H. Smith, P. Strenski, S. Swaney, F. Tanzi, S. Tsapepas, A. Tuminaro, P. Turgeon, G. Van Huben, S. Walker, L. Wang, D. Webber, C. Webb, B. Wile, P. Williams, B. Winter, T. Wohlfahrt, D. Wong, B. Wu, and F. Yee.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. C. W. Koburger III, W. F. Clark, J. W. Adkisson, E. Adler, P. E. Bakeman, A. S. Bergendahl, A. B. Botula, W. Chang, B. Davari, J. H. Givens, H. H. Hansen, S. J. Holmes, D. V. Horak, C. H. Lam, J. B. Lasky, S. E. Luce, R. W. Mann, G. L. Miles, J. S. Nakos, E. J. Nowak, G. Shahidi, Y. Taur, F. R. White, and M. R. Wordeman, "A Half-Micron CMOS Logic Generation," *IBM J. Res. Develop.* **39,** 215–227 (1995).
2. G. G. Shahidi, J. D. Warnock, J. Comfort, S. Fischer, P. A. McFarland, A. Acovic, T. I. Chappell, B. A. Chappell, T. H. Ning, C. J. Anderson, R. H. Dennard, J. Y.-C. Sun, M. R. Polcari, and B. Davari, "CMOS Scaling in the 0.1-μm, 1.X-Volt Regime for High-Performance Applications," *IBM J. Res. Develop.* **39,** 229–244 (1995).
3. C. Webb, C. Anderson, L. Sigal, K. Shepard, J. Liptay, J. Warnock, B. Curran, B. Krumm, M. Mayo, P. Camporese, E. Schwarz, M. Farrell, P. Restle, R. Averill, T. Slegel, W. Huott, Y. Chan, B. Wile, P. Emma, D. Beece, C. Chuang, and C. Price, "A 400 MHz S/390 Microprocessor," *ISSCC Digest of Technical Papers*, February 1997, pp. 168–169.
4. A. Pelella, P. Lu, Y. Chan, W. Huott, U. Bakhru, S. Kowalczyk, P. Patel, J. Rawlins, and P. Wu, "A 2ns Access, 500 MHz 288Kb SRAM Macro," *Digest of Technical Papers, Symposium on VLSI Circuits*, 1996, pp. 128–129.
5. K. L. Shepard, S. M. Carey, E. K. Cho, B. W. Curran, R. F. Hatch, D. E. Hoffman, S. A. McCabe, G. A. Northrop, and R. Seigler, "Design Methodology for the IBM S/390 Parallel Enterprise Server G4 Microprocessors," *IBM J. Res. Develop.* **41,** 515–547 (this issue, 1997).
6. W. Reohr, J. Navarro, Y. Chan, M. Mayo, B. Curran, B. Krumm, A. Pelella, P. Lu, U. Bakhru, S. Kowalczyk, J. Rawlins, S. Carey, and P. Wu, "Precharged Cache Hit Logic with Flexible Timing Control," *Digest of Technical Papers, Symposium on VLSI Circuits*, 1997, pp. 43–44.
7. S. Vassiliadis, M. Putrino, A. E. Huffman, B. J. Feal, and G. G. Pechanek, "Apparatus and Method for Prediction of Zero Arithmetic/Logic Results," U.S. Patent 4,947,359, 1991.

Received December 12, 1996; accepted for publication May 16, 1997

Leon Sigal *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (LJS at YKTVMV, ljs@vnet.ibm.com).* Mr. Sigal received a B.S. in biomedical engineering in 1985 from the University of Iowa and an M.S. in electrical engineering in 1986 from the University of Wisconsin at Madison. He worked at Hewlett-Packard's microprocessor development laboratory between 1986 and 1992. Mr. Sigal joined IBM in 1992 and has been leading the CMOS S/390 microprocessor circuit design interdivisional effort.

James D. Warnock *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (WARNOCK at YKTVMV, warnock@watson.ibm.com).* Dr. Warnock is a Research Staff Member in the VLSI Design Department at the IBM Thomas J. Watson Research Center. He received a B.S. degree in physics from the University of Ottawa, Ontario, Canada, in 1980 and a Ph.D. degree in physics from the Massachusetts Institute of Technology, Cambridge, MA, in 1985. He joined the IBM Research Division in Yorktown Heights in 1985, spending two years on solid-state physics research using femtosecond optical techniques. Since then he has spent time in research on silicon technology, focusing on advanced bipolar, CMOS, and biCMOS processes. He joined the VLSI design team in 1993 and is currently involved in CMOS circuit design for microprocessors.

Brian W. Curran *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (BCURRAN at PK705VMA, bcurran@vnet.ibm.com).* Mr. Curran received a B.S. degree in electrical engineering from the University of Wisconsin at Madison in 1984. He joined the IBM Data Systems Division (now System/390 Division) that same year. Mr. Curran has designed logic and circuits for large system processors and memory subsystems. He was a member of the teams which developed the System/3090 and ES/9121 processors. Most recently, he led the high-frequency effort of the S/390 G4 microprocessor. Mr. Curran has received several IBM Technical Achievement Awards. He has been issued seven patents relating to processor design and has three patents pending. Mr. Curran is currently a Senior Engineer in the Poughkeepsie Development Laboratory and a technical leader in the development of a future high-frequency full-custom CMOS microprocessor.

Yuen H. Chan *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (CHANY at PK705VMA, chany@vnet.ibm.com).* Mr. Chan is a Senior Engineer at the IBM Poughkeepsie S/390 Development Laboratory, working on custom VLSI circuit and SRAM designs. He joined IBM at the East Fishkill facility in 1977, and has worked on high-performance bipolar, biCMOS, and CMOS logic and array development. He received a B.S. in electrical engineering from Union College in 1977 and an M.S.E.E. from Syracuse University in 1984. Mr. Chan is a member of the IEEE. He has received an IBM Outstanding Technical Achievement Award and six IBM Invention Achievement Awards. He has authored and co-authored a number of technical papers, and is the holder of fifteen U.S. patents in circuit design.

Peter J. Camporese *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (PCAMP at PK705VMA, pcamporese@vnet.ibm.com).* Mr. Camporese is an Advisory Engineer in the Custom VLSI Design group for S/390 hardware development. He received a B.S. degree in electrical engineering from Polytechnic University in 1988, and an M.S. degree in computer engineering from Syracuse University in 1991. In 1988 he joined the IBM Data Systems Division, where he worked on system performance, circuit design, physical design, and VLSI integration. Most recently, he has been a technical team leader for the physical design and integration of the S/390 G4 CMOS microprocessor.

Mark D. Mayo *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (MAYOM at MHV, mayom@vnet.ibm.com).* Mr. Mayo is an Advisory Engineer in the S/390 Hardware Development Laboratory, Poughkeepsie. He joined IBM at the East Fishkill facility in 1980 and has since worked on the design of bipolar and CMOS gate arrays, standard cells, and custom microprocessor circuits. He is currently the circuit design leader for the S/390 buffer control element (BCE). Mr. Mayo received a B.S.E.E. with high honors from Lehigh University in 1980, and an M.S.E.E. from Syracuse University in 1985.

William V. Huott *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (ace@vnet.ibm.com).* Mr. Huott is an Advisory Engineer in S/390 custom microprocessor design. He received a B.S. in electrical engineering from Syracuse University in 1984. Mr. Huott is a design engineer responsible for the definition and design of the memory array built-in self-test (ABIST) circuitry and interfaces for high-frequency custom microprocessors. He is also responsible for the engineering test debug of high-frequency custom macros and support chips. Mr. Huott joined IBM in 1983 at the East Fishkill facility designing, testing, and debugging high-speed logic gate arrays and memory arrays. He holds four U.S. patents and several publications.

Daniel R. Knebel *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (KNEBELD at YKTVMV, knebel@vnet.ibm.com).* Mr. Knebel is an Advisory Engineer in the VLSI Test and Manufacturability Department at the IBM Thomas J. Watson Research Center. He joined the IBM Data Systems Division in 1982 and has since worked on high-performance processor technologies. He received a B.S. degree in electrical engineering from Purdue University in 1982. In 1988 Mr. Knebel received an IBM Outstanding Technical Achievement Award for his work on bipolar circuit technology.

Ching-Te Chuang *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (CCHUANG at YKTVMV, cchuang@watson.ibm.com).* Dr. Chuang is a Research Staff Member and Manager of the High-Performance Circuit group at the IBM Thomas J. Watson Research Center. He received a B.S.E.E. from the National Taiwan University, Taipei, Taiwan, in 1975 and a Ph.D. degree in electrical engineering from the University of California at Berkeley in 1982. From 1977 to 1982 he was a research assistant in the Electronics Research Laboratory, University of California, Berkeley, working on bulk and surface acoustic wave devices. He joined the Bipolar Devices and Circuits group at the IBM Thomas J. Watson Research Center in 1982, working on scaled bipolar

devices, technology, and circuits. He studied the scaling properties of epitaxial Schottky barrier diodes, carried out pioneering work on the perimeter effects of advanced double-poly self-aligned bipolar transistors, and designed the first subnanosecond 5Kb bipolar ECL SRAM. From 1986 to 1988, he was manager of the Bipolar VLSI Design group, working on low-power bipolar circuits, high-speed, high-density bipolar SRAMs, multi-Gb/s fiber-optic datalink circuits, and scaling aspects of bipolar/biCMOS devices and circuits. Since 1988, he has managed the High-Performance Circuit group, investigating high-performance logic and memory circuits. Since 1993, his group has worked primarily on the circuit design of high-performance CMOS microprocessors. Dr. Chuang served on the Device Technology Program Committee for IEDM in 1986 and 1987, and the Program Committee for the Symposium on VLSI Circuits from 1992 to 1997. He was Publication/Publicity Chairman for the Symposium on VLSI Technology and the Symposium on VLSI Circuits in 1993 and 1994. He was elected an IEEE Fellow in 1994 "for contributions to high-performance bipolar devices, circuits, and technology." Dr. Chuang is a member of the New York Academy of Science. He holds six U.S. patents and has authored or coauthored more than 110 papers.

James P. Eckhardt *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (ECKHARDT at PK705VMA).* Dr. Eckhardt is an Advisory Engineer in Advanced Circuit Development in the Poughkeepsie S/390 Design Center. He joined IBM in 1984 with an M.S. in electrical engineering from the Georgia Institute of Technology, and has since received his Ph.D. in electrical engineering, with emphasis on biCMOS circuitry, from the same institution. Dr. Eckhardt has written many papers, and received an IBM Outstanding Technical Achievement Award in 1996.

Philip T. Wu *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (PTWU at PK705VMA, ptwu@pk705vma.vnet.ibm.com).* Mr. Wu is a Senior Engineering Manager at the IBM Poughkeepsie S/390 Development Laboratory. He received a B.S. degree in electrical engineering from the University of Michigan in 1974, an M.S. degree in electrical engineering from Stanford University in 1976, and an M.B.A. degree from Rensselaer Polytechnic Institute in 1994. He joined IBM in 1976 and has since worked in advanced VLSI development. He is a member of the Project Management Institute (PMI). Mr. Wu has received three IBM Invention Achievement Awards and is the holder of seven U.S. patents in circuit design.